

Remarks

Entry of the amendments, reconsideration of the application, as amended, and allowance of all pending claims are respectfully requested. Upon entry of the amendments, claims 1-34 are pending.

With the above amendments, applicants are more particularly pointing out and distinctly claiming aspects of applicants' invention. Support for the amendments can be found throughout the specification (e.g., page 6, paragraph 18, lines 11-14; page 6, paragraph 20; page 10, paragraph 26, lines 6-10; page 10, paragraph 27, lines 7-9; and page 10, paragraph 28, lines 3-8). Thus, no new matter is being added.

In an effort to advance prosecution of this case and prior to discussing the particular rejections, applicants would like to take this opportunity to describe aspects of the present invention. This is for clarity purposes.

Applicants' invention is directed, in one or more aspects, to targeted searching for text of a computer program. The search is targeted at specific items, such that only the specific items are searched. Further, the search may be targeted at specific locations that are to be searched for the items. The specific items and/or locations to be searched are designated by the computer program itself, rather than by a user. Further, the items and/or locations to be searched are dynamically defined by the program, in that the search is dynamically revised based on information obtained during the search. By targeting the search, the chances of obtaining false positives are reduced and the search is less time consuming.

The targeted search is further described with the following example:

Sample Program - source.c

```
#include<myclass.h>
#include<stdio.h>
```

```
int main(){
print ("Hello World\n");
VariableX = 17;
FunctionCallY();
}
```

Sample Header file - myclass.h

```
Int VariableX = 5;

Int FunctionCallY()
{
printf ("HI There!");
}
```

Environment Variable

INCLUDE=C:\IBMCPPE\INCLUDE;C:\CLASSLIB\INCLUDE;E:\RICK\INCLUDE;

Files in Current Directory

Source.c
Myclass.h
Junk.h
Stdio.h
Miscfile.c
Miscfile.h

Filestructure

C:\junk
C:\ibmcpp\include
C:\ibmcpp\include\sys
C:\classlib\include
D:\miscellaneous
E:\rick\include
.
.
.

Illustrated above is a sample program, in which a function, FunctionCallY, is called by this program, but not defined by it. Likewise the variable, VariableX, is set by this program, but not defined by it. To learn more about this function or variable, one or more files are searched. The search begins with the source.c file and continues with one or more header files (e.g., other files pulled in by the #include statement in C and C++ languages) associated with the program.

This example focuses on the FunctionCallY function. The search begins with the source.c module, which is searched, along with any files identified in the #include statements. Further, if those files contain #include statements, then they also would be searched. In this example, it finds FunctionCallY in the myclass.h file. Files not associated with the program (e.g., not the source of the program, or not referenced by the program or files associated with the program) are

not searched. Thus, in this example, files junk.h, miscfile.c and miscfile.h, which are in the current directory, but not associated with this particular program, are not searched. Thus, the search described by applicants is targeted and the targeted items and/or locations to be searched are defined by the program itself, rather than a user. As described below, this is very different from Burk, the cited reference.

In the Office Action, dated April 9, 2004, claims 2, 4, 10-12, 14, 16, 20, 22, 24, 26 and 32 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicants regard as the invention. Applicants respectfully traverse this rejection, since each claimed feature specified in the Office Action does particularly point out and distinctly claim subject matter applicants regard as the invention, as described herein.

For example, claims 2, 10, 12, 20, 22, 24 and 32 now recite that the dynamically defining comprises using one or more rules of a programming language to select the one or more items to be searched. As claimed and described herein, the rules of a programming language are used to select the one or more items to be searched. This is described, for instance, at paragraph 20, page 6 of applicants' specification. For example, in the C or C++ programming language, items may be referenced by a #include statement. Thus, with a C or C++ program, a search for #include statement is performed. This is further shown in the sample program described above, as well as in the sample program of FIG. 3. The use of the rules to select the items to be searched allows the computer program to dynamically define the search, as claimed. Thus, applicants respectfully submit that the claims are definite in that they do particularly point out and distinctly claim subject matter which applicants regard as the invention.

Claims 4, 14 and 26 indicate that the one or more items searched comprise one or more classes. This is also described in the specification. For example, in paragraphs 30 and 33, the specification describes that for JAVA programs, as an example, the items searched for particular text include classes. Again, applicants respectfully submit that this is definite and describes one or more aspects of applicants' claimed invention.

As a further example, claims 10, 20, 22 and 32 now recite dynamically defining by a computer program one or more items in which to search for text of the computer program, the

one or more items being associated with the computer program. Again, applicants respectfully submit that this is definite and that it particularly points out and distinctly claims subject matter which applicants regard as the invention. This claimed element specifically describes that the computer program itself defines the items in which to search for text of the computer program. This is described above and throughout the specification. Thus, applicants respectfully submit that these claims are definite.

Further, claims 10, 20, 22 and 32 recite identifying by the computer program one or more locations in which the one or more items to be searched for the text are to included. Again, this is described throughout the specification and in the example above. This is part of applicants' claimed invention in which a targeted search is to be provided, so that locations that are not identified are not searched.

Based on the foregoing, applicants respectfully request withdrawal of the §112 rejection. Should the Examiner maintain this rejection, applicants respectfully request the Examiner to specifically indicate why the claims are indefinite.

In addition to the above, claims 1-32 are rejected under 35 U.S.C. 102(b) as being anticipated by Burk et al. (UNIX System Administrator's Edition", 1997). Applicants respectfully, but most strenuously, traverse this rejection to any extent deemed applicable to the amended claims.

In one aspect, applicants' invention is directed to targeted searching in which a particular search is targeted at specific items and/or at specific locations to be searched. The specific items and/or locations to be searched are designated by the computer program itself, rather than by a user. As one example, one or more rules specific to the programming language are used in selecting the items and/or locations to be searched. Additionally, the search is dynamically defined by the computer program, as the search progresses.

In one particular example, applicants claim a method of facilitating searching for text of computer programs. The method includes, for instance, identifying text of a computer program to be searched; and dynamically defining by the computer program one or more items associated with the computer program in which to search for the text. Thus, in this aspect of applicants'

claimed invention, the computer program dynamically defines the items in which to search for the identified text. This is very different from Burk.

While Burk describes Find and Grep commands, which are used to search files for selected text, the Find and Grep commands are very different from applicants' claimed invention. For instance, those commands require a user to specify the search criteria, which is quite different from applicants' claimed invention, in which it is the computer program itself that dynamically defines where the search is to be performed. Take the following sample C or C++ program:

SAMPLE PROGRAM

```
#INCLUDE<myclass.h>
#include<stdio.h>

INT MAIN () {
PRINT ("hello world\n");
VARIABLE X = 17;
FUNCTIONCALLY=();
}
```

As one example, assume the sample program is executed on a machine having the following environment variable:

```
INCLUDE=C:\IBMCPP\INCLUDE;C:\CLASSLIB\INCLUDE;
E:\RICK\INCLUDE;
```

In this example, the identified text (e.g., FunctionCally) is only searched, in accordance with an aspect of the present invention, in the selected files (e.g., source.c, myclass.h and stdio.h) in the following chosen directories: C:\IBMCPP\INCLUDE; C:\CLASSLIB\INCLUDE; and E:\RICK\INCLUDE. Other files and/or directories are not searched.

The search is defined by the #include statements of the computer program, rather than by a user specifying search criteria. By having the computer program define the search, the search may be revised (e.g., expanded) to include other targeted items or locations to be searched, as the search progresses. This is very different from Burk, as well as Grep and Find commands.

Grep and Find allow user defined arguments to be specified to help limit the scope of what is searched, such that the results are narrowed. However, Grep and Find have no way of dynamically defining (e.g., enlarging) the scope of the search by the computer program itself, as in applicants' claimed invention. Because the Grep and Find tools have no real concept of what they are searching, other than it being raw data, Grep and Find start with a large amount of objects to search and use arguments to limit which objects are searched. In contrast, applicants' claimed invention starts with a small set of objects and expands it dynamically to encompass other small sets through understanding of where to look based on knowledge of the object it is searching. This dynamic defining is by the computer program itself.

If Find or Grep was used to search on a method or function, the search would result in many incorrect matches, because neither Find nor Grep understands the difference between a function/method call and a function/method declaration or even a function/method prototype. Furthermore, if Find or Grep was used to find documentation on the same function/method, it would not understand the man pages or help files or pdfs or html to know how to find just the place that it is documented (no concept of indexes). It would show all of the references to function/method, as well as any supporting documentation. This is very different from applicants' claimed invention, in which the search is targeted and dynamically defined by the program to reduce false positives.

Based on the foregoing, applicants respectfully submit that independent claim 1, as well as the other independent claims, are patentable for the reasons herein. Further, the dependent claims are patentable for the same reasons as the independent claims, as well as for their own additional features.

Should the Examiner wish to discuss this case with applicants' attorney, please contact applicants' attorney at the below listed number.

Respectfully submitted,

Blanche E. Schiller
Blanche E. Schiller
Attorney for Applicants
Registration No.: 35,670

Dated: July 2, 2004.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579